



Ai Programming Assignment (1): River Crossing

Δημήτρης Λουρίδας
Χρήστος Γιαπιτζάκης

Εισαγωγή:

Το παρόν έντυπο παρέχει μια σύντομη περιγραφή του προγράμματος, των δυνατοτήτων του, της αρχιτεκτονικής του, των χρησιμοποιούμενων μεθόδων Τεχνητής Νοημοσύνης καθώς και παραδειγμάτων χρήσης και πειραματικών αποτελεσμάτων. Το πρόγραμμά μας αναπτύχθηκε για την επίλυση ενός προβλήματος διέλευσης μιας οικογένειας από τη μία όχθη ενός ποταμού στην άλλη. Η διέλευση γίνεται με ένα περιορισμένο χρονικό όριο, και ο στόχος είναι να βρεθεί ο βέλτιστος τρόπος να γίνει αυτή η διαδικασία. Χρησιμοποιείται ο αλγόριθμος A* για την εύρεση της βέλτιστης διαδρομής.

Περιγραφή Προβλήματος:

Το πρόγραμμα αναπτύχθηκε για την επίλυση του προβλήματος της μεταφοράς μιας οικογένειας από τη μία πλευρά του ποταμού στην άλλη, χρησιμοποιώντας διάφορες μεθόδους τεχνητής νοημοσύνης. Στο πρόβλημα αυτό μια οικογένεια πρέπει να διασχίσει ένα ποτάμι περπατώντας πάνω σε έναν κορμό που συνδέει τις δύο όχθες. Ο κορμός αντέχει το πολύ δύο άτομα κάθε φορά. Επίσης, η οικογένεια έχει μία μόνο λάμπα, την οποία πρέπει να έχει μαζί του ένα από τα άτομα που περπατούν κάθε φορά πάνω στον κορμό. Κάθε μέλος της οικογένειας χρειάζεται διαφορετικό χρόνο για να διασχίσει το ποτάμι περπατώντας πάνω στον κορμό. Ο χρόνος αυτός είναι σταθερός (προς οποιαδήποτε κατεύθυνση) για κάθε ένα από τα μέλη της οικογένειας. Όταν διασχίζουν το ποτάμι δύο μέλη της οικογένειας μαζί, ο χρόνος της διάσχησης είναι εκείνος του βραδύτερου. Ο χρόνος που χρειάζεται κάθε μέλος της οικογένειας για να διασχίσει το ποτάμι είναι γνωστός. Το πρόγραμμά μας βρίσκει τη βέλτιστη λύση, δηλαδή με ποια σειρά (και με ποια ζεύγη ή ατομικά σε κάθε μετακίνηση) πρέπει να κινηθούν τα μέλη της οικογένειας, ώστε να περάσει όλη η οικογένεια απέναντι στον ελάχιστο χρόνο.



Αρχιτεκτονική Προγράμματος:

Η αρχιτεκτονική του προγράμματος αποτελείται από 3 βασικές κλάσεις: την **FamilyMember** και την **State**, καθώς και από μια κλάση **Solver** που υλοποιεί τον αλγόριθμο A*. Επιπλέον, υπάρχει μια κύρια κλάση **Main** που χειρίζεται την εκκίνηση του προγράμματος.

1. Κλάση **FamilyMember**

Η κλάση αυτή αναπαριστά τα μέλη της οικογένειας και περιλαμβάνει:

- Ένα πεδίο **crossingTime** που αναπαριστά το χρόνο διέλευσης του μέλους.
- Έναν προκαθορισμένο constructor για αρχικοποίηση.
- Έναν constructor που δέχεται τον χρόνο διέλευσης ως όρισμα.
- Μεθόδους **setCrossingTime** και **getCrossingTime** για την ορισμό και ανάγνωση του χρόνου διέλευσης αντίστοιχα.
- Την υλοποίηση του **Comparable** interface για τη σύγκριση βάσει του χρόνου διέλευσης.

2. Κλάση **State**

Η κλάση αυτή περιγράφει την κατάσταση του περιβάλλοντος κατά τη διάρκεια της διέλευσης και περιλαμβάνει:

- Τα πεδία **lamp**, **left**, **right**, **g**, και **f** που αναπαριστούν τη θέση του φωτός, τα μέλη στην αριστερή όχθη, τα μέλη στη δεξιά όχθη, το κόστος διέλευσης, και τη συνολική ευρετική συνάρτηση αντίστοιχα.
- Δύο static πεδία **all_min1** και **all_min2** που χρησιμοποιούνται για την αποθήκευση των δύο μικρότερων χρόνων διέλευσης σε όλα τα μέλη της οικογένειας.
- Δύο constructors, έναν προκαθορισμένο και έναν παραμετροποιημένο.
- Μεθόδους **get_children**, **combinations**, **find_f**, **getG**, **get_lamp**, **get_left**, **get_right**, **isTerminal**, **getF** για διάφορες λειτουργίες.
- Υλοποίηση του **Comparable** interface για σύγκριση βάσει της συνολικής ευρετικής συνάρτησης **f**.

Οι μέθοδοι της κλάσης **State** παίζουν κρίσιμο ρόλο στην επίλυση του προβλήματος διέλευσης της οικογένειας. Ας αναλύσουμε τις κυριότερες μεθόδους αυτής της κλάσης:

Μέθοδος **get_children()**

Αυτή η μέθοδος επιστρέφει μια λίστα με τα πιθανά παιδιά της τρέχουσας κατάστασης, λαμβάνοντας υπόψη τους κανόνες του προβλήματος.

1. **Καθορισμός της θέσης του φωτός:**
 - Πρώτα, η μέθοδος καθορίζει σε ποια πλευρά του ποταμού βρίσκεται το φως (**left** ή **right**).
2. **Δημιουργία παιδιών:**
 - Αν το φως είναι στα αριστερά, δημιουργεί παιδιά μεταφέροντας έναν ή δύο ανάλογα με τον αριθμό των μελών που υπάρχουν στα αριστερά.
 - Αν το φως είναι στα δεξιά, δημιουργεί παιδιά μεταφέροντας δύο μέλη.
3. **Υπολογισμός του κόστους (**g**):**
 - Για κάθε παιδί υπολογίζεται το κόστος (**g**) με βάση τον χρόνο διέλευσης του πιο αργού μέλους που μεταφέρεται.
4. **Επιστροφή των παιδιών:**
 - Τα παιδιά προστίθενται σε μια λίστα παιδιών, η οποία επιστρέφεται στο τέλος.

Μέθοδος **combinations()**

Αυτή η μέθοδος δημιουργεί όλο το δυνατό σύνολο συνδυασμών των μελών μιας λίστας. Χρησιμοποιείται στη μέθοδο **get_children** για να παράγει όλες τις πιθανές συνδυαστικές επιλογές μελών που μεταφέρονται από τη μία όχθη στην άλλη.

Μέθοδος **find_f()**

Αυτή η μέθοδος υπολογίζει την ευρετική συνάρτηση **f** για την τρέχουσα κατάσταση, η οποία χρησιμοποιείται στον αλγόριθμο A*.

1. **Προετοιμασία των λιστών:**
 - Οι χρόνοι διέλευσης των μελών ταξινομούνται σε φθίνουσα σειρά.
2. **Υπολογισμός βασικού κόστους (**h**):**
 - Επιλέγονται οι δύο μικρότεροι χρόνοι διέλευσης από κάθε πλευρά του ποταμού και προστίθενται στο βασικό κόστος (**h**).
3. **Εξαρτημένος υπολογισμός κόστους ανά πλευρά:**
 - Ανάλογα με την πλευρά του ποταμού όπου βρίσκεται το φως, υπολογίζεται επιπλέον κόστος.
4. **Ενημέρωση του συνολικού κόστους (**f**):**
 - Το συνολικό κόστος (**f**) υπολογίζεται ως άθροισμα του βασικού κόστους (**h**) και του εξαρτημένου κόστους ανά πλευρά.

Μέθοδος **isTerminal()**

Αυτή η μέθοδος επιστρέφει **true** αν η τρέχουσα κατάσταση είναι τερματική, δηλαδή όταν όλα τα μέλη έχουν διασχίσει με επιτυχία τη γέφυρα και βρίσκονται στην άλλη όχθη.

Μέθοδος **bubbleSort()**

Αυτή η μέθοδος υλοποιεί τον αλγόριθμο ταξινόμησης φυσαλίδας (bubble sort). Χρησιμοποιείται για την ταξινόμηση των χρόνων διέλευσης μελών.

3. Κλάση **Solver**

Η κλάση αυτή υλοποιεί τον αλγόριθμο A* και περιλαμβάνει τη μέθοδο **searcher**, η οποία εκτελεί τον αλγόριθμο αναζήτησης, και χρησιμοποιείται για την εξαγωγή της βέλτιστης διαδρομής.

!Η συγκεκριμένη μέθοδος αναλύεται πιο παρακάτω

4. Κλάση **Main**

Στην κύρια κλάση γίνεται η είσοδος των δεδομένων από τον χρήστη και η εκκίνηση του αλγορίθμου. Επίσης, εκτυπώνονται τα αποτελέσματα, περιλαμβανομένου του χρόνου εκτέλεσης και της βέλτιστης διαδρομής.

Δυνατότητες

- Υποστηρίζει τον υπολογισμό του βέλτιστου μονοπατιού για τη μεταφορά της οικογένειας.
- Εφαρμόζει τον αλγόριθμο A* για την εύρεση του βέλτιστου μονοπατιού.
- Υποστηρίζει προσαρμοστική εκτύπωση αποτελεσμάτων, όπως η κατάσταση της λάμπας, ο χρόνος διασχίσματος κάθε μέλους της οικογένειας, κ.ά.
- Το πρόγραμμα μας τεσταρίστηκε για διαφορετικές τιμές του N(3,5,10,25) και λειτουργεί πλήρως αποδοτικά με το execution time να είναι πάντα της τάξης msec

Μέθοδοι Τεχνητής Νοημοσύνης

Ο αλγόριθμος A* είναι ένας αλγόριθμος αναζήτησης που χρησιμοποιείται για την εύρεση του βέλτιστου μονοπατιού σε έναν χώρο καταστάσεων. Καταφέρει να συνδυάσει την πλήρη αναζήτηση με την χρήση μιας ευρετικής συνάρτησης για να κατευθύνει την αναζήτηση προς τον στόχο, επιχειρώντας να ελαχιστοποιήσει το συνολικό κόστος.

Ο αλγόριθμος υλοποιείται στην κλάση Solver στην μέθοδο searcher και ξεκινάει με την καθορισμένη αρχική και τελική κατάσταση. Δημιουργεί μια λίστα, την **front**, που περιέχει τις καταστάσεις που πρέπει να αξιολογηθούν. Στη συνέχεια, επαναλαμβάνει τα εξής βήματα: επιλογή του κόμβου με το ελάχιστο κόστος από τη λίστα, αξιολόγηση του κόμβου, και δημιουργία των παιδιών του για εξέταση.

Η ευρετική συνάρτηση (heuristic) υπολογίζεται στη μέθοδο **find_f()** της κλάσης **State**. Σε γενικές γραμμές, η ευρετική συνάρτηση αξιολογεί το πόσο κοντά είναι μια κατάσταση στον στόχο. Χρησιμοποιείται ως πρόσθετη εκτίμηση του κόστους από τον τρέχοντα κόμβο μέχρι τον στόχο. Στην προκειμένη περίπτωση, η **find_f()** αναλαμβάνει τον υπολογισμό της ευρετικής συνάρτησης που χρησιμοποιείται στην εκτέλεση του αλγορίθμου A*.

Οι χαρακτηριστικές ιδιότητες του αλγορίθμου A* περιλαμβάνουν την πληρότητά του, καθώς εξερευνά όλους τους δυνατούς κόμβους, και την βελτιστότητά του, καθώς εγγυάται την εύρεση του βέλτιστου μονοπατιού. Η χρονική και χωρική πολυπλοκότητα είναι εκθετικές, εξαρτώμενες από τον αριθμό των καταστάσεων. Συνοπτικά, ο A* είναι ένας ισχυρός αλγόριθμος αναζήτησης που συνδυάζει αποτελεσματικά πλήρη και ευρετική αναζήτηση για επίλυση προβλημάτων μονοπατιού.

Παραδείγματα Χρήσης

Για να χρησιμοποιήσετε το πρόγραμμα, ακολουθήστε τα παρακάτω βήματα:

1. Καθορίστε τον αριθμό των μελών της οικογένειας.
2. Καθορίστε τον χρόνο διασχίσματος κάθε μέλους.
3. Καθορίστε το χρονικό όριο για το διάστημα διασχίσματος.

Μετά την εκτέλεση, το πρόγραμμα θα εμφανίσει το βέλτιστο μονοπάτι.

Πειραματικά Αποτελέσματα για διάφορα N:

1)

```
C:\Users\stell\OneDrive\Υπολογιστής\Erg>java Main
How many are the members of the family?
5
What is the crossing time of FamilyMember 1?
1
What is the crossing time of FamilyMember 2?
3
What is the crossing time of FamilyMember 3?
6
What is the crossing time of FamilyMember 4?
8
What is the crossing time of FamilyMember 5?
12
What is the crossing time limit?
30
Execution time: 1 msec
the lamp is on the right side
The left side after the change has:
12
8
6
3
1
-----
the lamp is on the left side
The left side after the change has:
3
1
The right side after the change has:
12
8
6
-----
```

```
the lamp is on the right side
The left side after the change has:
3
The right side after the change has:
12
8
6
1
-----
the lamp is on the left side
The left side after the change has:
12
8
3
The right side after the change has:
6
1
-----
the lamp is on the right side
The left side after the change has:
12
8
The right side after the change has:
6
3
1
-----
the lamp is on the left side
The left side after the change has:
12
8
3
1
The right side after the change has:
6
-----
```

```
the lamp is on the right side
The left side after the change has:
12
8
3
The right side after the change has:
6
1
-----
the lamp is on the left side
The left side after the change has:
12
8
6
3
1
The right side after the change has:
-----
Crossing time : 29
```

Σε περίπτωση που το time limit δεν αρκεί:

```
C:\Users\stell\OneDrive\Υπολογιστής\Erg>java Main
How many are the members of the family?
5
What is the crossing time of FamilyMember 1?
1
What is the crossing time of FamilyMember 2?
3
What is the crossing time of FamilyMember 3?
6
What is the crossing time of FamilyMember 4?
8
What is the crossing time of FamilyMember 5?
12
What is the crossing time limit?
28
Execution time: 1 msec
Solution not found
Crossing time : 29
```

2) Για $N=25$

```
-----  
the lamp is on the left side  
The left side after the change has:  
25  
24  
23  
22  
21  
20  
19  
18  
17  
16  
15  
14  
13  
12  
11  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1  
  
The right side after the change has:  
-----  
Crossing time : 231
```

!Παραλείπεται το μονοπάτι